We claim:

1. A method of compiling a netlist description of a logic design for programming into a hardware logic emulation system, the netlist description comprising combinational logic gates, sequential logic gates, data paths and clock paths, the sequential logic gates comprising flip-flops and latches, each of the flip-flops comprising a data input, a clock inputs and an output, the method comprising:

compiling the netlist description to create an emulation netlist, said compiling step comprising:

identifying every flip-flop in the emulation netlist;

changing the emulation netlist such that an adjustable delay element is disposed at the data input of each of the flip-flops of the netlist description; and

after said compiling step, setting a delay for said adjustable delay element to a value that eliminates the possibility of a hold time violation.

2. The method of claim 1 wherein said adjustable delay comprises a first flip-flop and a second flip flop, wherein said first flip-flop has an input, an output and a clock input, said second flip-flop has an input, an output and a clock input, said output of said first flip-flop in communication with said input of said second flip-flop.

3. The method of claim 2 wherein said delay is established in said adjustable delay element by varying frequencies input to said clock input on said first flip-flop and to said clock input on said second flip-flop.

4. A method processing a netlist description of a logic design for programming into an emulation system that eliminates hold time violations, the netlist description comprising combinational logic gates, sequential logic gates, data paths and clock paths, the sequential logic gates comprising flip-flops and latches, each of the flip-flops comprising a data input, a clock inputs and an output, the emulation system comprised of programmable logic chips interconnected together, the method comprising:

compiling the netlist description to create an emulation netlist, said compiling step comprising inserting an adjustable delay element at the data input of each of the flip-flops of the netlist description;

calculating data path delay time and clock path delay time, the clock paths and data paths may be passing through multiple of the programmable logic chips;

calculating clock skew value between a pair of flip-flops; and

setting a delay value for said adjustable delay element that makes said data path delay greater than said clock skew.

5. The method of claim 4 wherein said adjustable delay comprises a first flip-flop and a second flip flop, wherein said first flip-flop has an input, an output and a clock input, said second flip-flop has an input, an output and a clock input, said output of said first flip-flop in communication with said input of said second flip-flop.

6. The method of claim 5 wherein said delay is established in said adjustable delay element by varying frequencies input to said clock input on said first flip-flop and to said clock input on said second flip-flop.

7. The method of claim 4 further comprising removing selected ones of said adjustable delay elements from the netlist description where said data path delay already greater than said clock skew without setting said delay value.